

**APPLICATION
FOR
UNITED STATES LETTERS PATENT**

APPLICANT NAME: Jana H. Jenkins

TITLE: COMPUTER-IMPLEMENTED METHOD, SYSTEM AND
PROGRAM PRODUCT FOR REVIEWING A MESSAGE
ASSOCIATED WITH COMPUTER PROGRAM CODE

DOCKET NO.: RSW920030213US1

INTERNATIONAL BUSINESS MACHINES CORPORATION

CERTIFICATE OF MAILING UNDER 37 CFR 1.10

I hereby certify that, on the date shown below, this correspondence is being deposited with the United States Postal Service in an envelope addressed to the Commissioner for Patents, Box Patent Application, Washington, D.C. 20231 as "Express Mail Post Office to Addressee" Mailing Label No. EV225574751US

on October 9, 2003

Dorothea Rubbone
Name of person mailing paper

Dorothea Rubbone 10/09/2003
Signature Date

**COMPUTER-IMPLEMENTED METHOD, SYSTEM AND PROGRAM
PRODUCT FOR REVIEWING A MESSAGE ASSOCIATED WITH COMPUTER
PROGRAM CODE**

Background of the Invention

1. Field of the Invention

[0001] The present invention generally relates to a computer-implemented method, system and program product for reviewing a message associated with computer program code. Specifically, the present invention provides uniformity in messages that exist to support computer program code.

2. Related Art

[0002] As computers continue to become more integrated in everyday life, computer software continues to become more sophisticated. For example, today a computer user can prepare a tax return, pay bills, purchase goods or services, etc., from the comfort of his/her personal computer. Such convenience is provided by various software packages. To this extent, computer program developers are constantly working to improve existing products, or introduce new products to the market.

[0003] In developing computer program code, a developer will usually include certain "messages" designed to support the product for the end-user. Typical messages that can be provided include error messages, warning messages and information messages.

Although each of these messages serves a different purpose, they are usually necessary to

fully support the program code being developed. Unfortunately, there is currently no uniformity in providing such messages. For example, a message for a particular purpose generated by developer "A" might include one line of text, while the same message generated by developer "B" might include four lines of text. Still yet, different developers might utilize different terminology or definitions when generating messages. This could be especially problematic when a single product is developed by multiple developers. Accordingly, there is currently no way to accurately predict the length or content of most messages.

[0004] As a result of this predicament, program integrated information (PII) counts are usually inaccurate and highly inflated to try to accommodate some unknown number.

This raises various concerns since translation between languages is often necessary for a program. Specifically, it is common for a software product to be made available in multiple countries/regions of the world. As such, lingual translation of interfaces, messages and the like is necessary. The longer the messages that are utilized to support a program, the greater the translation costs. Thus, with no real control or uniformity over message development, translation costs for a program can be substantial and unpredictable.

[0005] In view of the foregoing, there exists a need for a computer-implemented method, system and program product for reviewing a message associated with computer program code. Specifically, a need exists for a system that can be configured for a particular review standard. A further need exists for a message to be reviewed for errors based on

the review standard. Still yet, a need exists to clearly present any errors detected during the review process, and provide an opportunity to address the errors.

Summary of the Invention

[0006] In general, the present invention provides a computer-implemented method, system and program product for reviewing a message associated with computer program code. Specifically, under the present invention, a standard for reviewing the message is first configured. The review standard can be based on a stored resource containing review parameters, manually designated review parameters, or both. When a message is created or edited, the message is reviewed based on the review standards. Any errors that are detected are presented to provide an opportunity for correction. The system also allows messages to be deleted and corresponding notifications to be communicated to appropriate parties.

[0007] A first aspect of the present invention provides a computer-implemented method for reviewing a message associated with computer program code, comprising: providing the message; configuring a review standard for reviewing the message; reviewing the message based on the review standard to detect errors; displaying any errors that are detected concurrently with the message; and revising the message to address the errors.

[0008] A second aspect of the present invention provides a computerized system for reviewing a message associated with computer program code, comprising: a message configuration system for configuring the message; a standard configuration system for configuring a review standard for reviewing the message; and a message analysis system

for reviewing the message based on the review standard to detect any errors, and for displaying the errors if detected.

[0009] A third aspect of the present invention provides a program product stored on a recordable medium for reviewing a message associated with computer program code, which when executed, comprises: program code for configuring the message; program code for configuring a review standard for reviewing the message; and program code for reviewing the message based on the review standard to detect any errors, and for displaying the errors if detected.

[0010] Therefore, the present invention provides a computer-implemented method, system and program product for reviewing a message associated with computer program code.

Brief Description of the Drawings

[0011] These and other features of this invention will be more readily understood from the following detailed description of the various aspects of the invention taken in conjunction with the accompanying drawings in which:

[0012] Fig. 1 depicts a system for reviewing a message associated with computer program code according to the present invention.

[0013] Fig. 2 depicts the message review system of Fig. 1 in greater detail.

[0014] Fig. 3A depicts the message creation system of Fig. 2. in greater detail.

[0015] Fig. 3B depicts the message creation interface as generated by the message creation system of Fig. 3A.

[0016] Fig. 4A depicts the message edit system of Fig. 2 in greater detail.

[0017] Fig. 4B depicts the message edit interface as generated by the message creation system of Fig. 4A.

[0018] Fig. 5A depicts the message deletion system of Fig. 2 in greater detail.

[0019] Fig. 5B depicts the message deletion interface as generated by the message creation system of Fig. 5A.

[0020] Fig. 6 depicts a flow diagram of a method for reviewing a message associated with computer program code according to the present invention.

[0021] Fig. 7 depicts the message creation routine of Fig. 6.

[0022] Fig. 8 depicts the message edit routine of Fig. 6.

[0023] Fig. 9 depicts the message deletion routine of Fig. 6.

[0024] Fig. 10 depicts the message review routine of Fig. 6.

[0025] The drawings are merely schematic representations, not intended to portray specific parameters of the invention. The drawings are intended to depict only typical embodiments of the invention, and therefore should not be considered as limiting the scope of the invention. In the drawings, like numbering represents like elements.

Detailed Description of the Invention

[0026] As indicated above, the present invention provides a computer-implemented method, system and program product for reviewing a message associated with computer program code. Specifically, under the present invention, a standard for reviewing the message is first configured. The review standard can be based on a stored resource

containing review parameters, manually designated review parameters, or both. When a message is created or edited, the message is reviewed based on the review standards. Any errors that are detected are presented to provide an opportunity for correction. The system also allows messages to be deleted and corresponding notifications to be communicated to appropriate parties.

[0027] Referring now to Fig. 1, a system 10 for reviewing a message associated with computer program code is shown. As depicted, system 10 includes a computer system 12 that receives computer program code 14 (hereinafter simply referred to as “program code 14”) and messages 16 as developed by one or more developers 15. As will be further described below, message review system 18 is an executable program or the like that analyzes messages 16 based upon message review standards to determine whether any errors are present therein. Any errors that are present could be displayed to the developers 15 (and concurrently with the message and/or associated program code) in a result interface 32. This provides the developers 15 with an opportunity to address the errors.

[0028] It should be understood that computer system 12 is intended to represent any type of computerized device capable of executing programs and performing the functions described herein. For example, computer system 12 could be a personal computer, a handheld device, a workstation, client, server, etc. In addition, computer system 12 could be implemented as a stand-alone system, or as part of a computerized network such as the Internet, local area network (LAN), wide area network (WAN), virtual private network (VPN), etc. In the case of the former, program code 14 and messages 16 could be

provided to computer system 12 on a recordable medium such as a diskette or CD-ROM (e.g., by a developer 15). In the case of the latter, program code 14 and messages 16 could be made available to computer system 12 over the network (e.g., via e-mail, file sharing, etc.). Furthermore, if implemented within a network, computer system 12 could represent a client or a server. As known, communication between clients and a server could occur via a direct hardwired connection (e.g., serial port), or via an addressable connection that may utilize any combination of wireline and/or wireless transmission methods. The server and the clients may utilize conventional network connectivity, such as Token Ring, Ethernet, WiFi or other conventional communications standards. Moreover, connectivity could be provided by conventional TCP/IP sockets-based protocol. In this instance, the clients could utilize an Internet service provider to establish connectivity to the server.

[0029] Regardless of its implementation, as shown computer system 12 comprises central processing unit (CPU) 20, memory 22, bus 24, input/output (I/O) interfaces 26, external devices/resources 28 and storage unit 30. CPU 20 may comprise a single processing unit, or be distributed across one or more processing units in one or more locations, e.g., on a client and computer system. Memory 22 may comprise any known type of data storage and/or transmission media, including magnetic media, optical media, random access memory (RAM), read-only memory (ROM), a data cache, etc. Moreover, similar to CPU 20, memory 22 may reside at a single physical location, comprising one or more types of data storage, or be distributed across a plurality of physical systems in various forms.

[0030] I/O interfaces 26 may comprise any system for exchanging information to/from an external source. External devices/resources 28 may comprise any known type of external device, including speakers, a CRT, LCD screen, handheld device, keyboard, mouse, voice recognition system, speech output system, printer, monitor/display, facsimile, pager, etc. Bus 24 provides a communication link between each of the components in computer system 12 and likewise may comprise any known type of transmission link, including electrical, optical, wireless, etc.

[0031] Storage unit 30 can be any system (e.g., database) capable of providing storage for information under the present invention. Such information could include, for example, program code, messages, stored resources such as rules, policies and the like containing message review parameters, files containing results of message analysis, etc. As such, storage unit 30 could include one or more storage devices, such as a magnetic disk drive or an optical disk drive. In another embodiment, storage unit 30 includes data distributed across, for example, a local area network (LAN), wide area network (WAN) or a storage area network (SAN) (not shown). Furthermore, although not shown, additional components, such as cache memory, communication systems, system software, etc., may be incorporated into computer system 12.

[0032] Referring now to Fig. 2, the functions of message review system 18 will be described in greater detail. As depicted, message review system 18 includes standard configuration system 40, message configuration system 42, notification system 49 and message analysis system 50. Standard configuration system 40 provides the mechanism by which the standard for reviewing messages is defined. Specifically, standard

configuration system 40 could spawn a standards interface (not shown) or the like through which developers 15 (Fig. 1) can designate certain stored resources such as policies, rules, etc. (e.g., within storage unit 30) that contain the parameters for reviewing messages. Such parameters could include, for example, character/line limits. Standard configuration system 40 could also allow for dictionaries, glossaries or other reference resources to be provided or designated. Still yet, standard configuration system 40 could allow developers 15 to manually define desired review parameters. Regardless of the scenario, standard configuration system 40 is used to define the standard by which messages 16 will be reviewed.

[0033] Message configuration system 42 provides the mechanism by which a message is provided for review under the present invention. To this extent, message configuration system 42 includes a message creation system 44 that allows new messages to be created for subsequent review under the present invention. Further, message configuration system 42 also includes a message edit system 46 that allows existing messages to be edited and subsequently reviewed under the present invention. A message deletion system 48 can also be provided within message configuration system 42 to allow existing messages to be deleted. In general, a message that is deleted need not be reviewed under the present invention. In any event, these systems will be further described below in conjunction with Figs. 3A-B, 4A-B and 5A-B.

[0034] Using the review standard that has been configured via standard configuration system 40, any provided messages 16 can be automatically reviewed by message analysis system 50. Specifically, message analysis system 50 will analyze messages 16 based on

the configured review standard, and output any errors in the result interface 32. For example, if a message went over the defined limit of lines or contained a spelling error, these would be indicated as errors in result interface 32. Using the result interface 32, developers 15 can then address the errors (e.g., delete text, correct the spelling error, etc.).

[0035] Referring now to Figs. 3A-B, the functions of message creation system 44 will be described in greater detail. As depicted, message creation system 44 includes creation interface system 52 and identifier assignment system 54. When a developer desires to create a new message, creation interface system 52 will spawn creation interface 56, which is shown in Fig. 3B. As shown, creation interface 56 includes check boxes (or the like) 58 for the developer to indicate whether he/she wishes to view creation interface 56 concurrently with the program code 14 associated with the new message being created. If so, the program code 14 can be displayed with creation interface 56 in a split screen or the like. To create a new message, the developer will input text into message input field 60. Message type boxes 62 allow the developer to designate whether the new message is an error or a warning message. If so, the developer could input an explanation for the error or warning message in explanation field 64, and a suggested response by the end-user in suggested response field 66. If the box is not checked, it could be presumed that the new message is an information message. Once all information has been input, the developer can select the submit button 68 to complete the creation process, or the cancel button 70 to cancel the creation process. It should be understood that although not shown, creation interface 56 could include "radio buttons" that allow the developer to select previously drafted template language for message input field 60, explanation field 64 and/or

suggested response field 66. Providing this option helps to further the message uniformity provided under the present invention.

[0036] If submit button 68 is selected, identifier assignment system 54 of Fig. 3A will assign a unique identifier to the new message, and then save the new message according to the unique identifier. Specifically, under the present invention all messages 16 (Fig. 1) are typically catalogued/stored according to a unique identifier. This allows the messages to be individually identified for future reference. Thereafter, notification system 49 (Fig.2) could optionally send notifications to a distribution list of appropriate parties such as other developers, supervisors, etc., as stored in storage unit 30 (Figs. 1 and 2). In any event, once the new message has been created, message analysis system 50 of Fig. 2 will analyze the new message as described above. In particular, the new message will be analyzed based on the configured review standard for any errors. If no errors are present, or once any errors are corrected, the message can be cleared for use with program code 14.

[0037] Referring now to Figs. 4A-B, the functions of message edit system 46 will be described in greater detail. As shown, message edit system 46 includes edit interface system 72, message retrieval system 74 and edit recording system 76. If a developer wishes to edit a message, edit interface system 72 will spawn edit interface 78 of Fig. 4B. Once spawned, the developer should indicate which existing message he/she wishes to edit. Accordingly, the developer will input the unique identifier corresponding to the desired message in identifier field 79. For example, if developer 15 wishes to edit message "A," he/she will input the unique identifier that corresponds to message "A."

Thereafter, message retrieval system 74 of Fig. 4A will use the unique identifier to locate and retrieve/obtain the message from storage. The content 84 of the message will be displayed for the developer in message field 82. Using message field 82, the developer can then make any desired changes to the message. Upon completion, the developer can select submit button 86 to finalize the edit process, or the cancel button 88 to cancel the edit process. If submit button 86 is selected, edit recording system 76 will record the edited message according to its unique identifier. Thereafter, notification system 49 (Fig. 2) could optionally send notifications to the distribution list, and message analysis system 50 (Fig. 2) will analyze the edited message based on the review standards to determine if any errors are present. As indicated above, any such errors can be represented to the developer in the review interface. As further shown in Fig. 4B, similar to creation interface 56 of Fig. 3B, edit interface 78 also includes check boxes for the developer to indicate whether he/she wishes to concurrently view program code 14 associated with the message being edited. If so, program code 14 can be displayed with edit interface 78 in a split screen or the like.

[0038] Referring now to Figs. 5A-B, the functions of message deletion system 48 of Fig. 2 will be described in greater detail. As depicted, message deletion system 48 includes deletion interface system 90 and message retrieval system 92. If a developer wishes to delete an existing message, deletion interface system 90 will spawn deletion interface 96 of Fig. 5B. Similar to edit interface 78 of Fig. 4B, deletion interface 96 includes an identifier field 97 for inputting the unique identifier corresponding to the message wished to be deleted, and a set of check boxes 98 to indicate whether the program code

associated with the message to be deleted should be displayed concurrently. Once the unique identifier is input, message retrieval system 92 will retrieve/obtain the corresponding message, and display the same in message field 100. Using buttons 102 or 104, the developer can then decide whether to delete the message or cancel the deletion process. If the deletion button 102 is selected, the message will be deleted and notification will be sent to any involved parties (e.g., supervisors, other developers, etc.) informing them of the deletion via notification system 49 (Fig. 2). Since the message has been deleted, there is little or no need for subsequent review by the message analysis system.

[0039] Referring now to Fig. 6, a method flow diagram according to the present invention is shown in greater detail. As depicted, the developer will first configure the review standard in step S1. As indicated above, configuration of the review standard can include designating certain stored resources containing review parameters, designating resources such as dictionaries, glossaries and the like, manually designating review parameters, or any combination thereof. In any event, messages can be provided for review based on the review standard. Specifically, in step S2, it is determined whether a new message(s) is being created. If so, the message creation routine is run in step S3. If not, it is determined in step S4 whether an existing message is being edited. If so, the message edit routine is run in step S5. If, however, a message is neither being created nor being edited, the message deletion routine can be run in step S6, notifications can be sent to a distribution (list) in step S8, and the process will end in step S9. Conversely, any messages resulting from the message creation routine of step S3 or message edit routine

of step S5 will be subject to the review routine of step S7 before notifications are sent to the distribution list in step S8 and the process is ended in step S9.

[0040] Referring now to Fig. 7, the message creation routine of step S3 is shown in greater detail. As depicted, in step C1, it is determined whether the program code associated with the new message should be displayed concurrently with the new message. If so, the program code can be displayed in a split screen in step C2. In step C3, new text for the message is entered (e.g., using the creation interface of Fig. 3B). Thereafter, it is determined whether the new message is an error or warning message in step C4. If not, the creation routine is ended in step C7. If, however, the new message is an error or warning message, the developer can enter an explanation in step C5 and a suggested action by the user in step C6 before the process is ended in step C7.

[0041] Referring now to Fig. 8, the message edit routine of step S5 of Fig. 6 is shown in greater detail. As depicted, in step E1, it is determined whether the program code associated with the existing message should be displayed concurrently with the existing message. If so, the program code can be displayed in a split screen in step E2. In step E3, the unique identifier/message number corresponding to the existing message is input and the message is obtained. Thereafter, the message is displayed in step E4 (e.g., in the edit interface of Fig. 4B) for editing in step E5. Once the editing is complete the process is ended in step E6.

[0042] Referring now to Fig. 9, the message deletion routine of step S6 of Fig. 6 is shown in greater detail. As depicted, in step D1, it is determined whether the program code associated with the existing message should be displayed concurrently with the existing

message. If so, the program code can be displayed in a split screen in step D2. In step D3, the unique identifier/message number corresponding to the existing message is input and the message is obtained. Thereafter, the message is displayed in step D4 (e.g., in the deletion interface of Fig. 5B). In step D5, it is determine whether the message should be deleted. If not, the process is ended in step D8. If, however, the message should be deleted in step D5, the message is formally deleted in step D6. Thereafter, the process is ended in step D7.

[0043] Fig. 10 depicts the review routine of step S7 of Fig. 6 in greater detail. In step R1, the message is reviewed based on the configured review standard. In step R2, it is determined whether any errors are present. If not, the message is saved in step R3. If, however, errors were detected in step R2, they can be displayed for the developer (e.g., in the result interface of Fig. 1) in step R4. As indicated above, the errors can be displayed concurrently with the actual message and/or the associated program code in a split screen or the like. In any event, the errors can be revised in step R5 and the message will be reviewed again in step R1. This process will repeat until no errors are found so that the message can be saved in step R3 and the process ended In step R6.

[0044] It should be understood that the present invention can be realized in hardware, software, or a combination of hardware and software. Any kind of computer system(s) - or other apparatus adapted for carrying out the methods described herein - is suited. A typical combination of hardware and software could be a general purpose computer system with a computer program that, when loaded and executed, carries out the respective methods described herein. Alternatively, a specific use computer, containing

specialized hardware for carrying out one or more of the functional tasks of the invention, could be utilized. The present invention can also be embedded in a computer program product, which comprises all the respective features enabling the implementation of the methods described herein, and which - when loaded in a computer system - is able to carry out these methods. Computer program, software program, program, or software, in the present context mean any expression, in any language, code or notation, of a set of instructions intended to cause a system having an information processing capability to perform a particular function either directly or after either or both of the following: (a) conversion to another language, code or notation; and/or (b) reproduction in a different material form.

[0045] The foregoing description of the preferred embodiments of this invention has been presented for purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed, and obviously, many modifications and variations are possible. Such modifications and variations that may be apparent to a person skilled in the art are intended to be included within the scope of this invention as defined by the accompanying claims. For example, although not shown, notifications could be sent to a distribution list upon all message reviews, not only upon message deletion. In this case, notifications could be generated and sent by message analysis system 50 of Fig. 2.